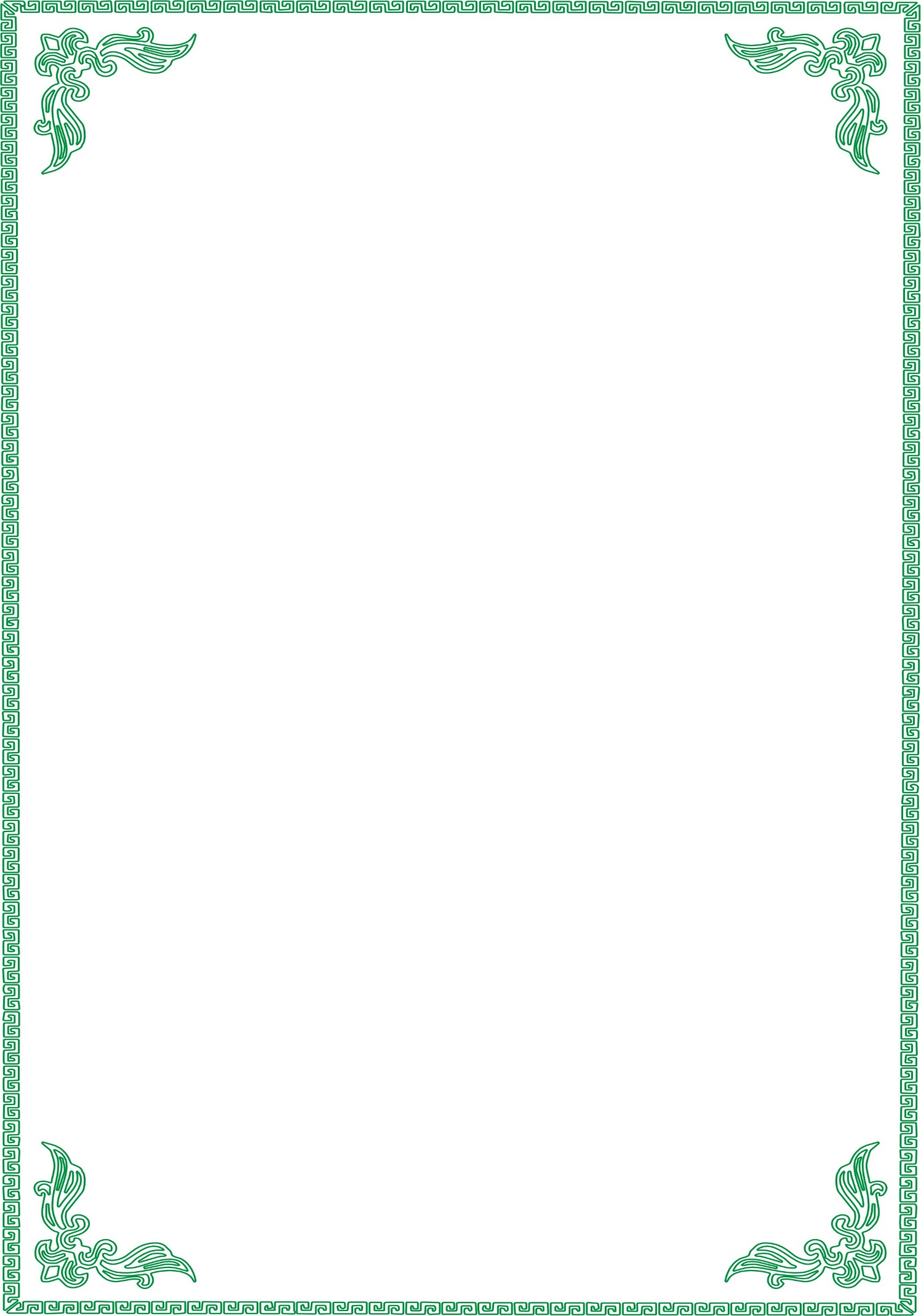
****

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**~~~\*\*\*\*\*\*\*\*\*~~~**

****

**BÁO CÁO**

**MÔN: THỰC TẬP CHUYÊN MÔN**

Đề tài: Quản lý điểm thi

***\*Giảng viên hướng dẫn:*** *Nguyễn Thu Hường*

***\*Sinh viên thực hiện:***Nguyễn Văn Nam (171200318)

*Lớp: CNTT 1 - Khóa: 58*

*Hà Nội, ngày 27 tháng 06 năm 2020*

MỤC LỤC

[I. **Giới thiệu chung** 3](#_Toc41569787)

[II. **Mô tả chương trình** 3](#_Toc41569788)

[1. Sơ đồ phân rã chức năng 3](#_Toc41569789)

[2. Mô hình thực thể liên kết 5](#_Toc41569790)

[3. Các chức năng của chương trình 5](#_Toc41569791)

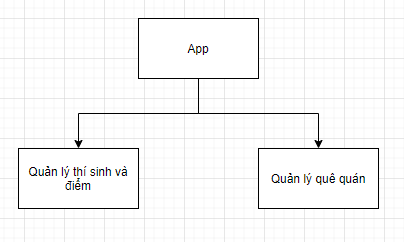
[III. **Tài liệu tham khảo** 21](#_Toc41569792)

1. **Giới thiệu chung**

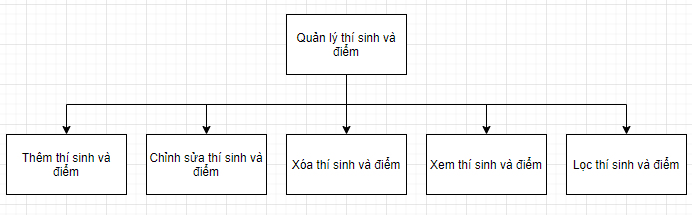
* Sự bùng nổ của ngành công nghệ thông tin trong những năm cuối thế kỷ 20 đầu thế kỷ 21 đã tác động đến mọi mặt của đời sống xã hội từ khoa học kỹ thuật đến kinh tế xã hội. Nhờ có tin học hoạt động sản xuất, quản lý trở nên phong phú, thuận tiện hơn, và mang lại nhiều lợi ích hơn. Công nghệ thông tin đã xâm nhập vào hầu hết các hoạt động của con người và ngày càng khẳng định vai trò của nó.
* Trong những năm gần đây, công nghệ thông tin ở nước ta đã có những tiến bộ vượt bậc. Nó không còn xa lạ với mọi người và ngày càng đước ứng dụng rộng rãi trong tất cả các lĩnh vực của đời sống xã hội cũng như trong công tác quản lý.
* Quản lý một hoạt động vô cùng rộng lớn cho dù ở bất kỳ cấp quản lý nào, nó không đơn giản chỉ ở cấp độ quản lý con người mà nó còn quản lý nhiều đối tượng khác như: quản lý sản xuất, quản lý kinh tế, quản lý xã hội…
* Trong trường học thì quản lý điểm thi là một trong những hoạt động quản lý chính và quan trọng nhất. Công tác quản lý điểm thi là một hoạt động không thể thiếu được đối với một trường học và ứng dụng tin học trong hoạt động quản lý điểm thi là một việc nên làm và hết sức cần thiết.
* Trong đề tài này em xin được trình bày quá trình xây dựng phần mềm quản lý điểm với các công việc chủ yếu: phân tích, thiết kế, lập trình hệ thống thông tin quản lý điểm thi.
* Hệ thống quản lý điểm thi cho phép có thể quản lý và thực hiện các hoạt động như: thêm, chỉnh sửa, xóa điểm thi của thí sinh, quản lý quê quán.

1. **Mô tả chương trình**
2. Sơ đồ phân rã chức năng

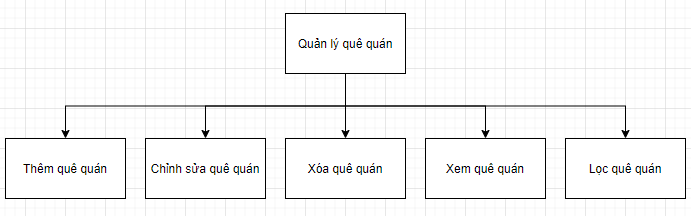
* App có thể chia thành 2 phần: Quản lý thí sinh và điểm, Quản lý quê quán.



* Phần Quản lý thí sinh và điểm gồm các chức năng: Thêm, sửa, xóa, xem chi tiết và tìm kiếm. Các chức năng này tác động đến SQL Server.

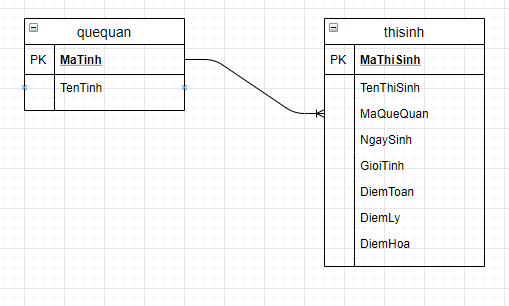


* Phần Quản lý quê quán gồm các chức năng: Thêm, sửa, xóa, xem chi tiết và tìm kiếm. Các chức năng này tác động đến SQL Server.



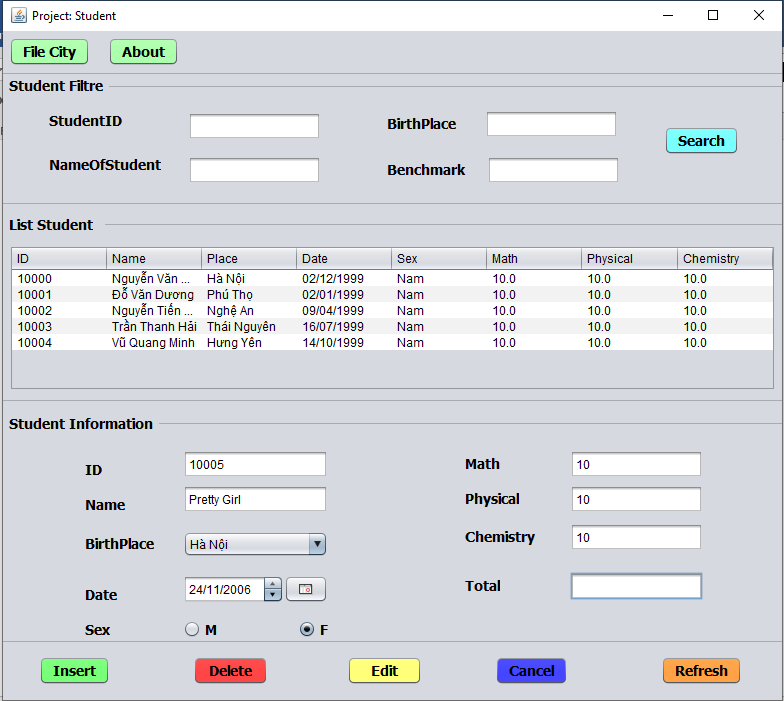
1. Mô hình thực thể liên kết

* App có 2 table: thisinh và quequan. Tương ứng để lưu thí sinh và quê quán.
* Table thisinh gồm các fields: mã thí sinh (tự động tăng), tên thí sinh, mã quê quán, ngày sinh, giới tính, điểm toán, điểm lý, điểm hóa.
* Table quequan gồm các fields: mã tỉnh (tự động tăng), tên tỉnh.

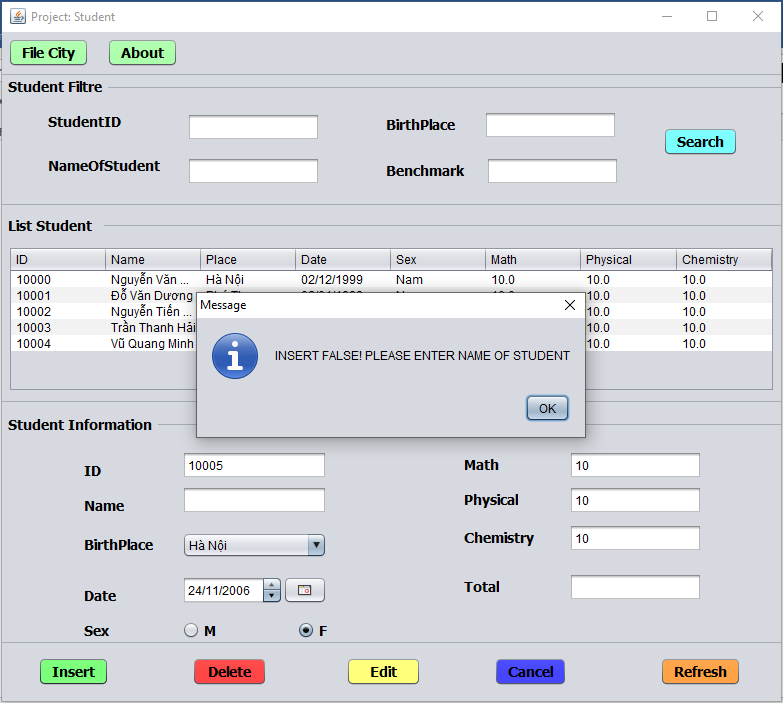


1. Các chức năng của chương trình

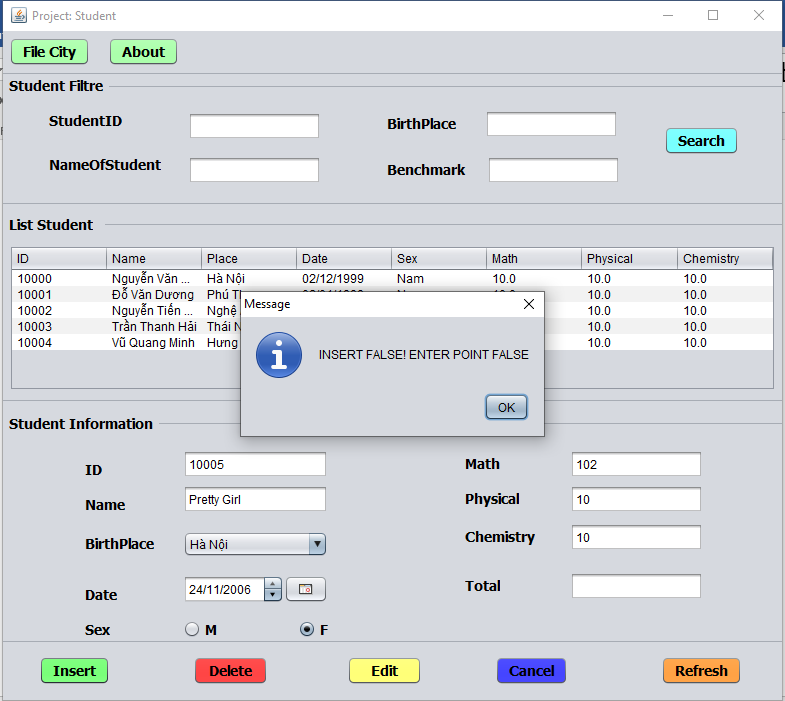
* Phần Quản lý thí sinh và điểm:
* Chức năng Thêm: Khi người dùng thêm thành công thì thí sinh mới sẽ được thêm vào SQL Server. Ngược lại, thông báo sẽ được gửi tới người dùng.
* Giao diện:
* Giao diện chính:



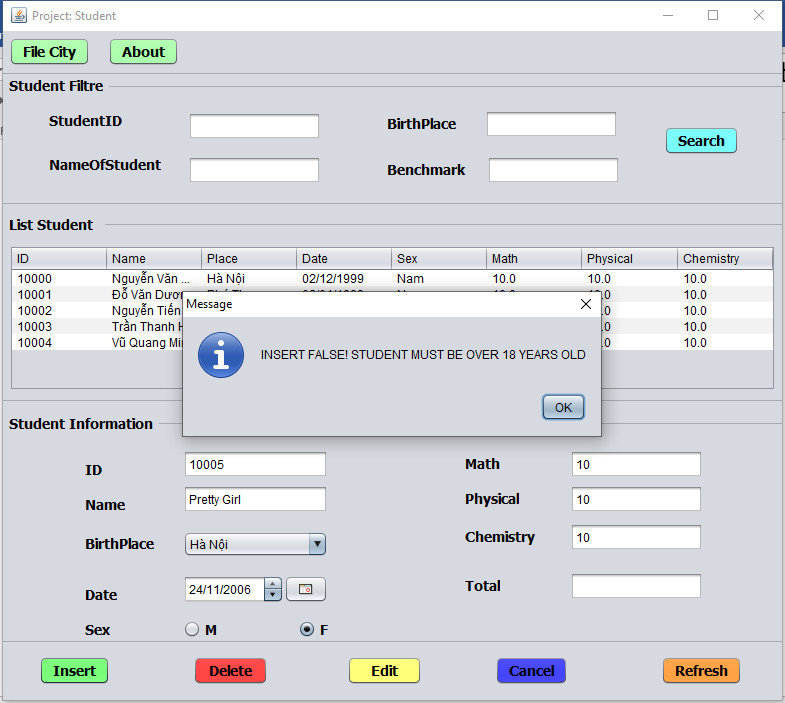
* Thông báo lỗi khi người dùng không nhập tên thí sinh:



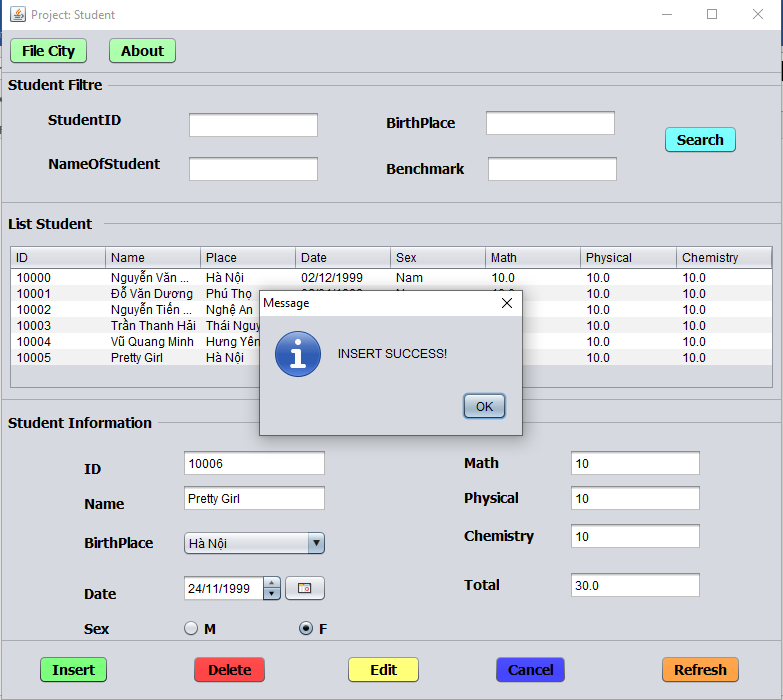
* Thông báo lỗi khi người dùng nhập sai hoặc không nhập điểm:



* Thông báo lỗi khi người dùng nhập ngày sinh của thí sinh dưới 18 tuổi:



* Thông báo thêm thành công khi người dùng nhập đúng và đủ dữ liệu:



* Code:

tableModel.setRowCount(0);

**for** (**int** i = 0; i < listStudent.size(); i++) {

showResult(i, listStudent.get(i));

}

//cÃ i Ä‘áº·t mÃ£ thÃ­ sinh

setIDStudent();

//kiá»ƒm tra danh sÃ¡ch tá»‰nh thÃ nh

**if** (listCity.isEmpty()) {

JOptionPane.*showMessageDialog*(rootPane, "INSERT FALSE! LIST CITY IS EMPTY");

} **else** {

//thÃªm thÃ­ sinh khi thÃ´ng tin Ä‘Ãºng theo yÃªu cáº§u

**if** (checkPoint() == **true** && checkName() == **true** && checkDate() == **true**) {

jTextFieldID.setText(MaTS + "");

ThiSinh thiSinh = **new** ThiSinh();

setStudent(thiSinh);

TinhTongDiem();

listStudent.add(thiSinh); //thÃªm vÃ o danh sÃ¡ch thÃ­ sinh

showResult(listStudent.size() - 1, thiSinh); //hiá»‡n thÃ­ sinh ra table

**new** ThiSinhSQL().addThiSinh(thiSinh); //thÃªm thÃ­ sinh vÃ o database

jTextFieldID.setText(MaTS + 1 + "");

JOptionPane.*showMessageDialog*(rootPane, "INSERT SUCCESS!");

} **else** **if** (checkName() == **false**) {

JOptionPane.*showMessageDialog*(rootPane, "INSERT FALSE! PLEASE ENTER NAME OF STUDENT");

} **else** **if** (checkPoint() == **false**) {

JOptionPane.*showMessageDialog*(rootPane, "INSERT FALSE! ENTER POINT FALSE");

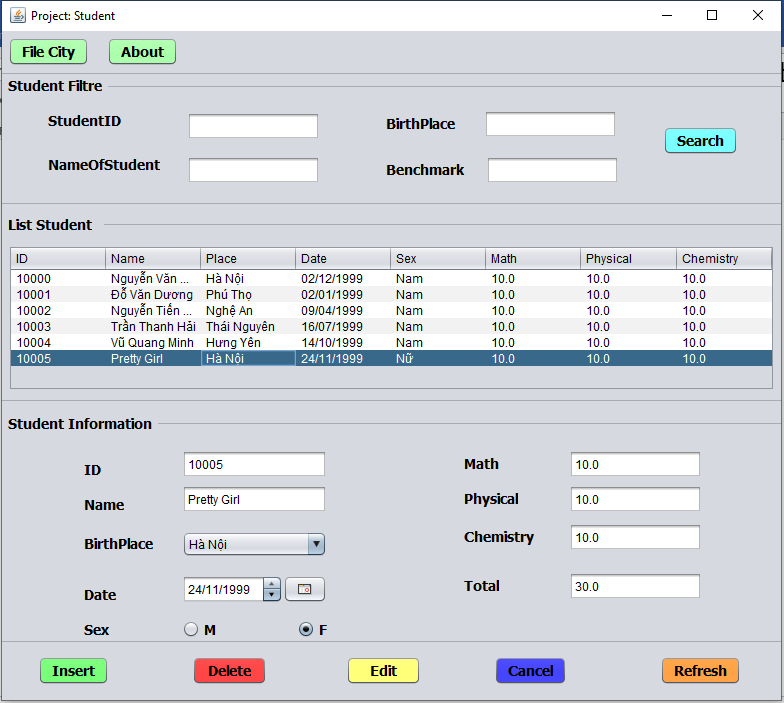
} **else** **if** (checkDate() == **false**) {

JOptionPane.*showMessageDialog*(rootPane, "INSERT FALSE! STUDENT MUST BE OVER 18 YEARS OLD");

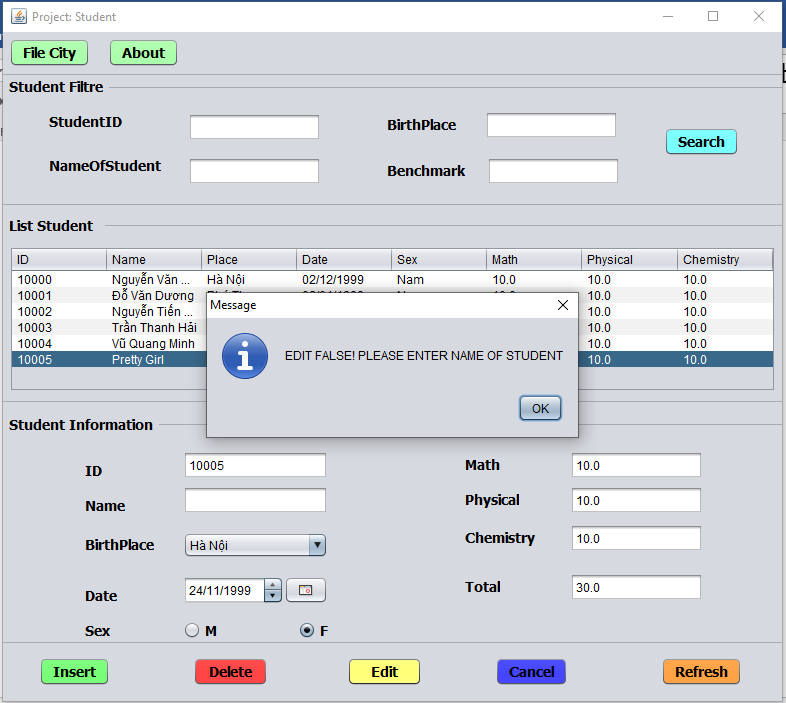
}

}

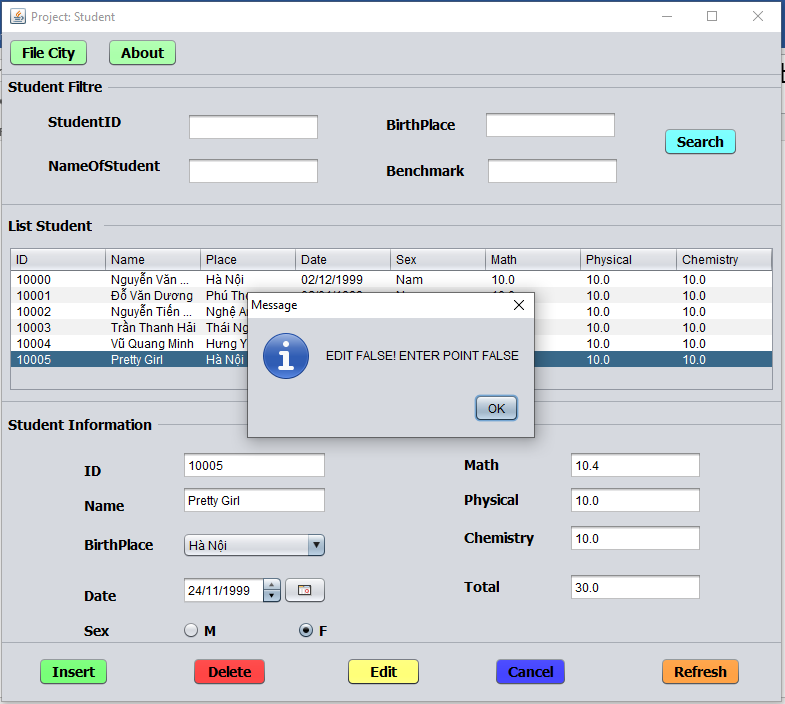
* Chức năng Sửa: Khi người dùng sửa thành công thì thí sinh sẽ được sửa lại và lưu vào SQL Server. Ngược lại, thông báo sẽ được gửi tới người dùng.
* Giao diện:
* Khi người dùng chọn 1 thí sinh trong table:



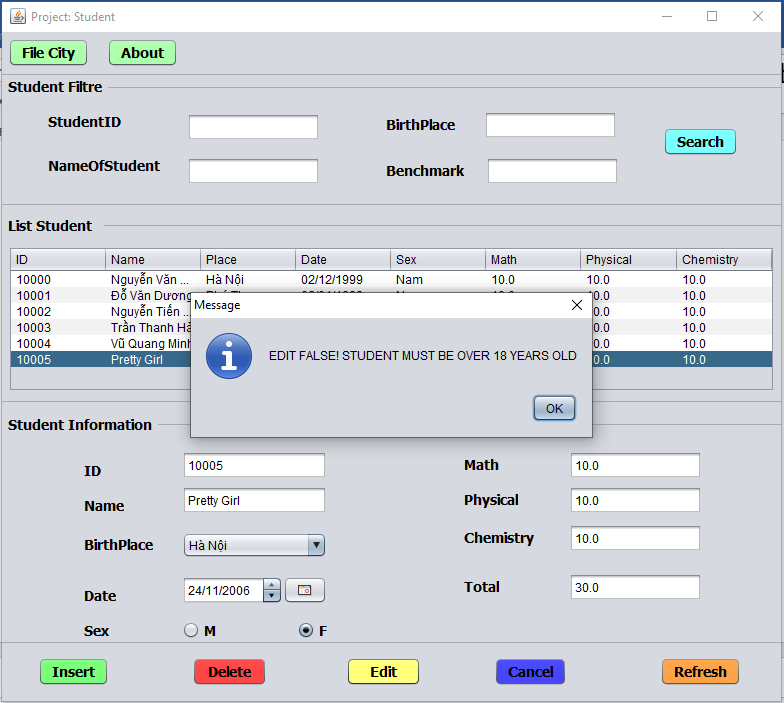
* Thông báo lỗi khi người dùng không nhập tên thí sinh:



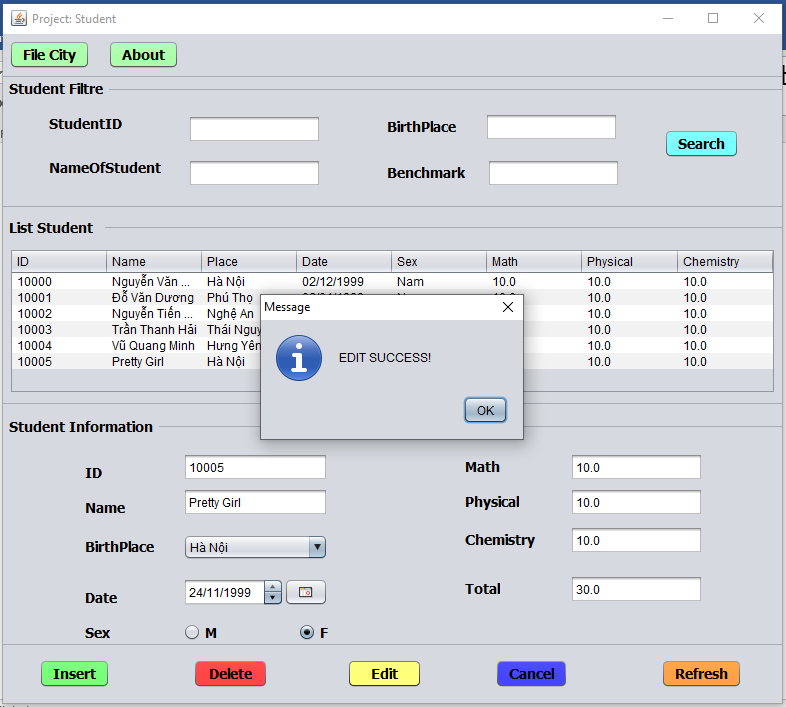
* Thông báo lỗi khi người dùng nhập sai hoặc không nhập điểm:



* Thông báo lỗi khi người dùng nhập ngày sinh của thí sinh dưới 18 tuổi:



* Thông báo sửa thành công khi người dùng nhập đúng và đủ dữ liệu:



* Code:

**int** index = jTableList\_Student.getSelectedRow(); //láº¥y vá»‹ trÃ­ Ä‘Ã£ chá»�n trong table

//sá»­a thÃ­ sinh khi thÃ´ng tin Ä‘Ãºng yÃªu cáº§u

**if** (index >= 0 && checkPoint() == **true** && checkName() == **true** && checkDate() == **true**) {

**int** id = Integer.*parseInt*(jTextFieldID.getText()); //láº¥y mÃ£ thÃ­ sinh Ä‘Ã£ chá»�n

**for** (**int** i = 0; i < listStudent.size(); i++) {

**if** (id == listStudent.get(i).getMaThiSinh()) {

setStudent(listStudent.get(i));

tableModel.removeRow(index);

showResult(index, listStudent.get(i));

TinhTongDiem();

**new** ThiSinhSQL().UpdateThiSinh(id, listStudent.get(i));

**break**;

}

}

JOptionPane.*showMessageDialog*(rootPane, "EDIT SUCCESS!");

} **else** **if** (checkName() == **false**) {

JOptionPane.*showMessageDialog*(rootPane, "EDIT FALSE! PLEASE ENTER NAME OF STUDENT");

} **else** **if** (checkPoint() == **false**) {

JOptionPane.*showMessageDialog*(rootPane, "EDIT FALSE! ENTER POINT FALSE");

} **else** **if** (checkDate() == **false**) {

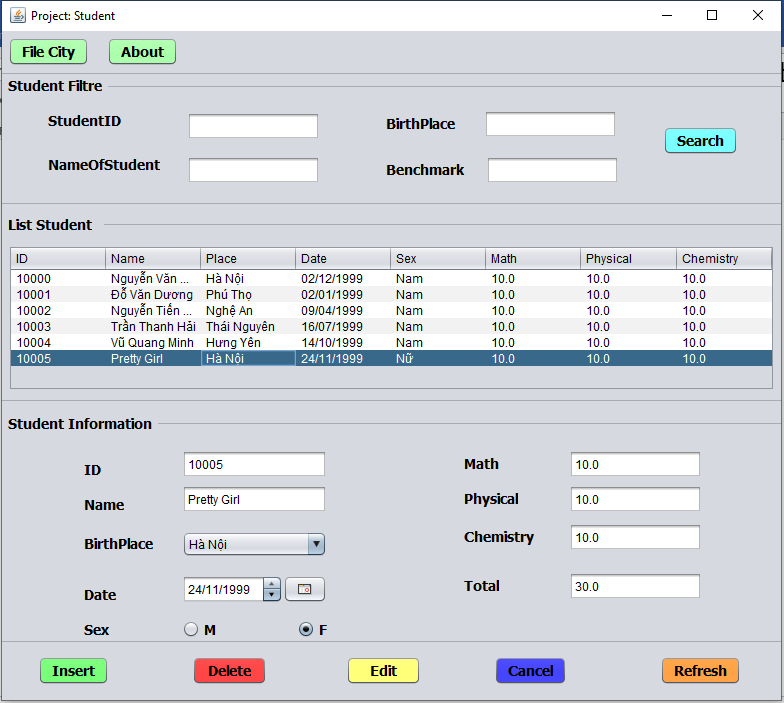
JOptionPane.*showMessageDialog*(rootPane, "EDIT FALSE! STUDENT MUST BE OVER 18 YEARS OLD");

} **else** {

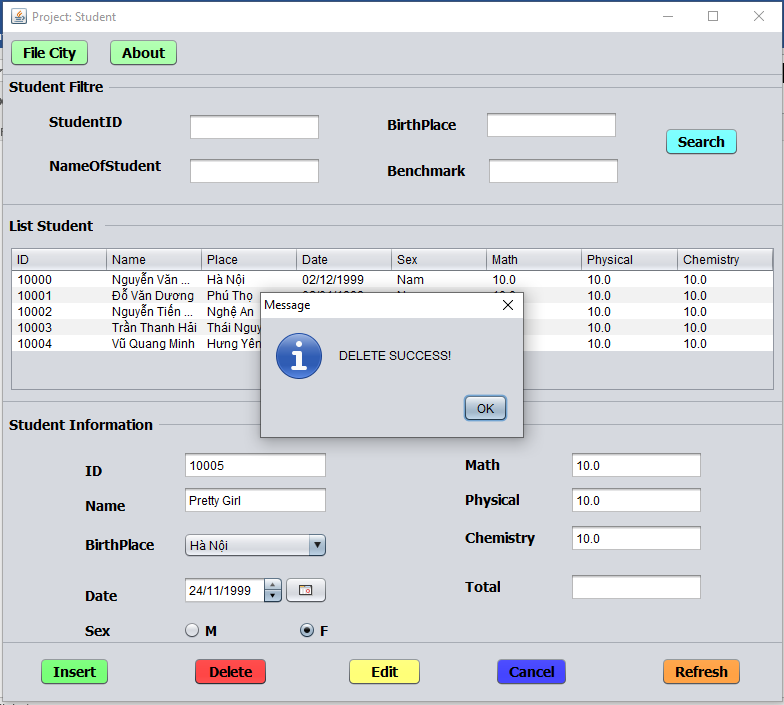
JOptionPane.*showMessageDialog*(rootPane, "EDIT FALSE! ID OF STUDENT DOES NOT EXIST!");

}

* Chức năng Xóa: nếu người chọn 1 thí sinh trong table và bấm xóa thì thí sinh đó sẽ được xóa trong SQL Server.
* Giao diện:
* Khi người dùng chọn 1 thí sinh trong table:



* Thông báo xóa thành công:



* Code:

**int** index = jTableList\_Student.getSelectedRow(); //láº¥y vá»‹ trÃ­ Ä‘Ã£ chá»�n trong table

//xÃ³a thÃ­ sinh

**if** (index >= 0) {

**for** (**int** i = 0; i < listStudent.size(); i++) {

**if** (Integer.*parseInt*(jTextFieldID.getText()) == listStudent.get(i).getMaThiSinh()) {

**new** ThiSinhSQL().DeleteThiSinh(listStudent.get(i));

**break**;

}

}

tableModel.removeRow(index);

jTextFieldTotal.setText("");

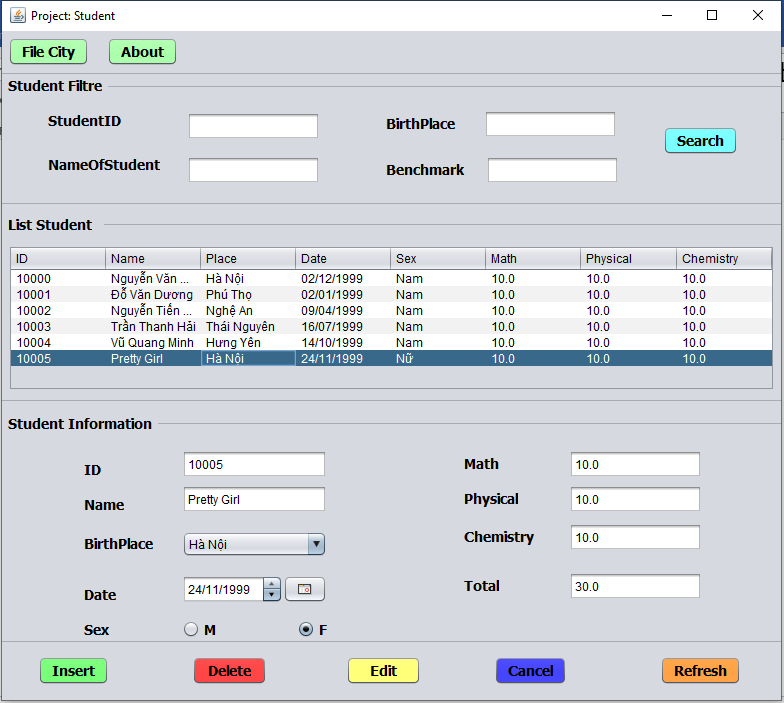
JOptionPane.*showMessageDialog*(rootPane, "DELETE SUCCESS!");

} **else** {

JOptionPane.*showMessageDialog*(rootPane, "DELETE FALSE! ID OF STUDENT DOES NOT EXIST!");

}

* Chức năng Xem chi tiết: khi người dùng chọn 1 thí sinh trong table thì sẽ hiển thị các thông tin của thí sinh cho người dùng.
* Giao diện:
* Hiển thị thông tin và tổng điểm của thí sinh:



* Code:

**int** index = jTableList\_Student.getSelectedRow();

jTextFieldID.setText(tableModel.getValueAt(index, 0).toString());

jTextFieldName.setText(tableModel.getValueAt(index, 1).toString());

jComboBoxPlace.setSelectedItem(tableModel.getValueAt(index, 2));

**try** {

jDateChooser.setDate(formatter.parse(tableModel.getValueAt(index, 3).toString()));

} **catch** (ParseException ex) {

Logger.*getLogger*(MainForm.**class**.getName()).log(Level.***SEVERE***, **null**, ex);

}

**if** (tableModel.getValueAt(index, 4).toString().equalsIgnoreCase("Nam")) {

jRadioButtonMale.setSelected(**true**);

} **else** {

jRadioButtonFemale.setSelected(**true**);

}

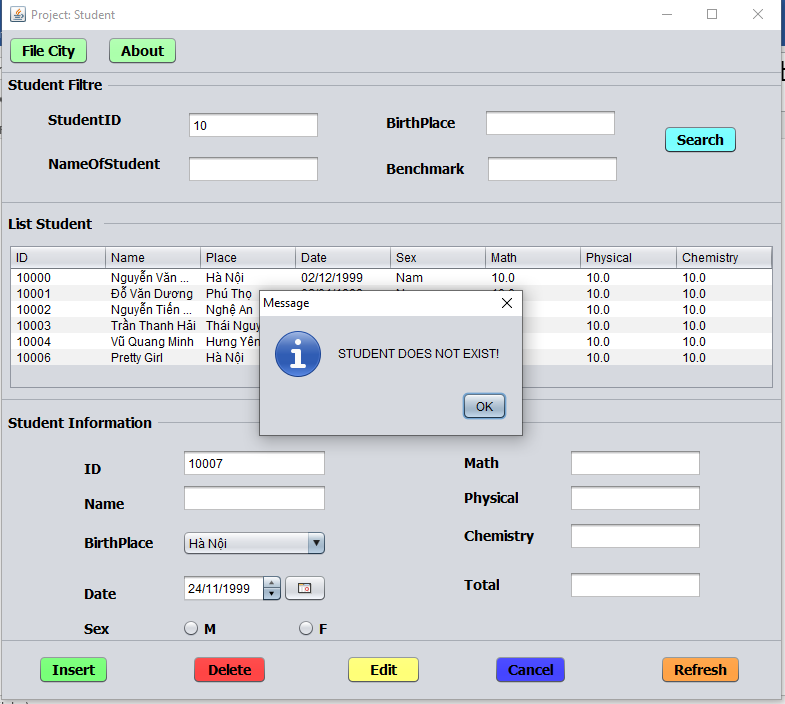
jTextFieldMath.setText(tableModel.getValueAt(index, 5).toString());

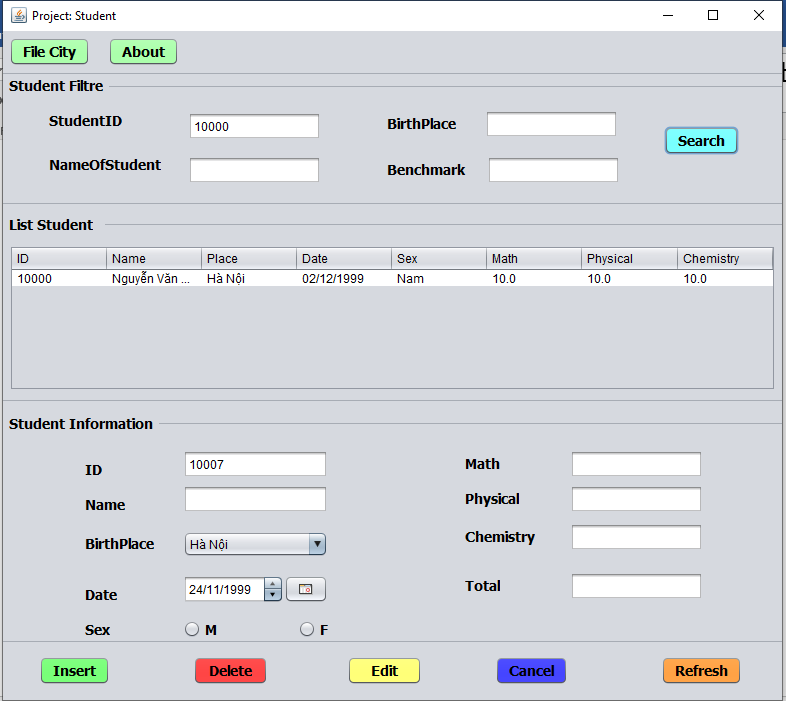
jTextFieldPhysical.setText(tableModel.getValueAt(index, 6).toString());

jTextFieldChemistry.setText(tableModel.getValueAt(index, 7).toString());

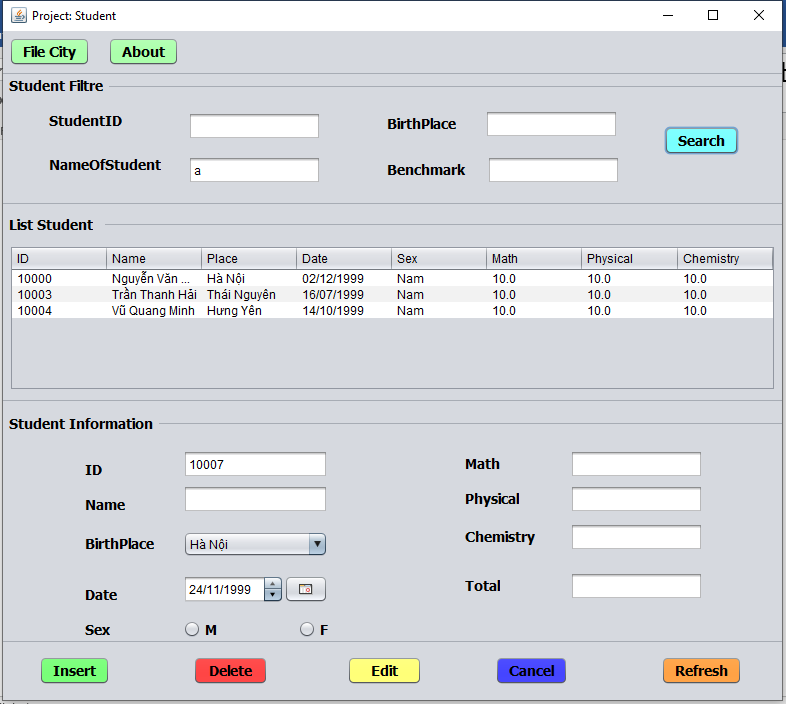
TinhTongDiem();

* Chức năng Lọc: hiển thị thí sinh có chứa mã thí sinh/tên thí sinh/tên tỉnh thành/điểm chuẩn mà người dùng nhập.
* Giao diện:
* Tìm kiếm theo mã thí sinh:

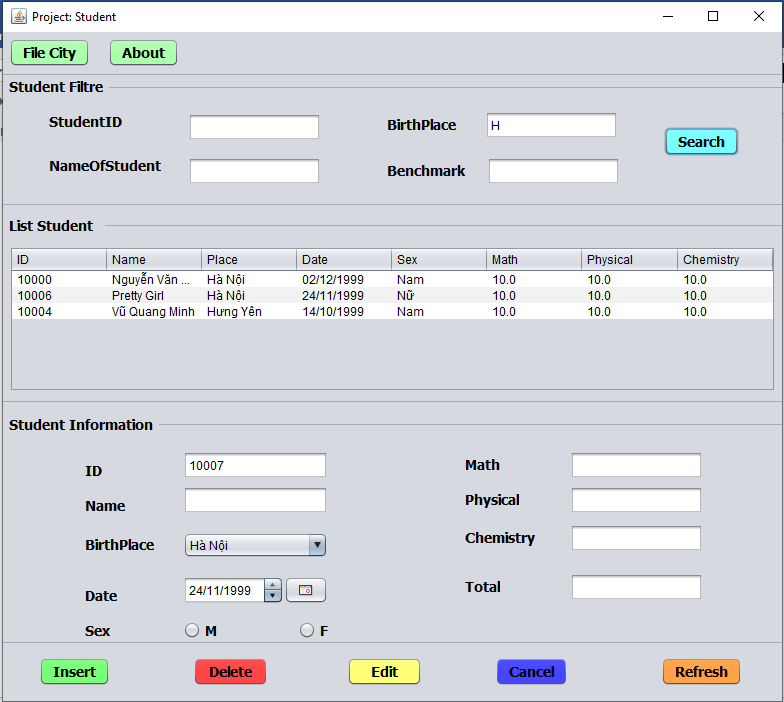




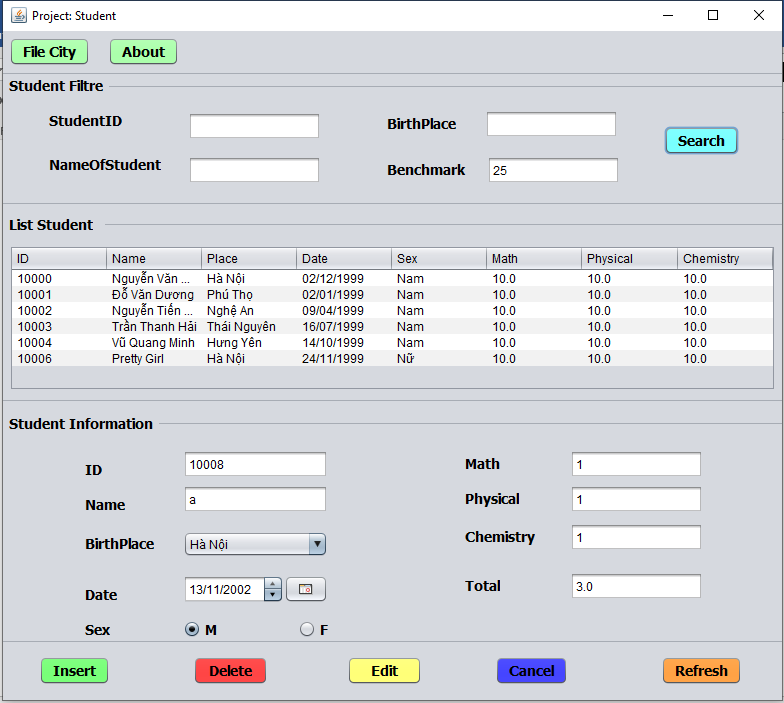
* Lọc theo tên thí sinh:



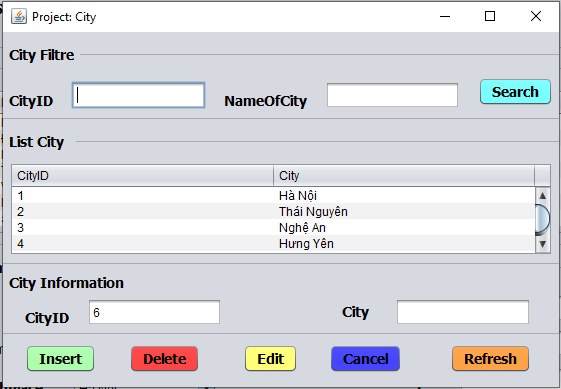
* Lọc theo tên tỉnh thành:



* Lọc theo điểm chuẩn, hiện ra những thí sinh có tổng điểm >= điểm chuẩn:



* Phần Quản lý quê quán: các chức năng và thông báo đến người dùng tương tự Phần Quản lý thí sinh và điểm.
* Giao diện:



1. **Tài liệu tham khảo**

* Google: <https://vietjack.com/java_swing/tong_quan_ve_java_swing.jsp>
* Youtube: <https://www.youtube.com/watch?v=tt0Rk70GS1Q&list=PLE1qPKuGSJaCmWIvdMmFovaYYxKlHIHAA>